IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BOARD OF PATENT APPEALS AND INTERFERENCES

In Re Application of: Yen-Kuang Chen et al. Application No. 10/612,542 Filed: July 1, 2003 For: BITSTREAM BUFFER MANIPULATION WITH A SIMD MERGE INSTRUCTION	Examiner: David H. Malzahn Art Unit: 2123 **CERTIFICATE OF TRANSMISSION** I hereby certify that this correspondence is being transmitted to the United States Patent and Trademark Office, or facsimile transmitted to Fax No. (571)273-8300 on 9-11-2007 /Lawrence M. Mennemeier/ Date Lawrence M. Mennemeier

APPELLANT'S BRIEF UNDER 37 CFR § 41.37
IN SUPPORT OF APPELLANT'S APPEAL TO THE BOARD OF PATENT APPEALS AND INTERFERENCES

Mail Stop Appeal Brief-Patents
Commissioner of Patents
PO Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Appellant hereby submits this Brief in support of an appeal from a final decision of the Examiner, in the above-referenced case. Appellant respectfully requests consideration of this appeal by the board of Patent Appeals and Interference for allowance of the above-referenced patent application.

TABLE OF CONTENTS

I. Real Party in Interest

The real party in interest in the present appeal is Intel Corporation of Santa Clara, California, the assignee of the present application.

II. Related Appeals and Interferences

There are no related appeals or interferences to appellant's knowledge that would have a bearing on any decision of the Board of Patent Appeals and Interferences.

III. Status of the Claims (independent claims shown in bold)

Claims **1**-39 are pending in the application.

Claims **1**-14, **15**-23 and **24**-29 and stand rejected under 35 USC § 101 as allegedly being directed to a program per se.

Claims **30**-39 are allowed.

Final rejection of claims **1**-14, **15**-23 and **24**-29 is being appealed.

IV. Status of Amendments

An official amendment and response to a first Office Action mailed 8/16/2006 was submitted by appellant on 2/16/2007 and was entered. A Final Office Action was mailed on 3/26/2007. A Notice of Appeal was transmitted on 7/11/2007, and an appeal ensued.

Accordingly, the claims stand as of appelant's response of 2/16/2007, and are reproduced in clean form in the Claims Appendix.

V. Summary of Claimed Subject Matter

Appellant's disclosure describes methods, apparatus, and program means for performing bitstream buffer manipulation with a SIMD (Single Instruction Multiple Data) merge instruction. The method of one embodiment comprises determining whether any unprocessed data bits for a partial variable length symbol exist in a first data block. A shift merge operation is performed to merge the unprocessed data bits from the first data block with a second data block. A merged data block is formed. A merged variable length symbol comprised of the unprocessed data bits and a plurality of data bits from the second data block is then extracted from the merged data block.

Claim 1, for example, sets forth a method comprising: determining whether any unprocessed data bits for a partial variable length symbol exist[1] in a first data block[2]; and

[1] "But if the determination at block **840** is that no whole symbol is present in the loaded data, a check is made at block **842** as to whether a partial piece of a symbol is located within the unprocessed bits. If the finding at block **842** is that no partial piece of a symbol is present in the unprocessed bits, meaning that the remaining bits may be invalid or unrecognized, the flow proceeds to tab E **846**." **(US 2005/0108312 A1 p. 12, par. 92, lines 8-15; Fig. 8C, 842)**

[2] "A check is made at block **822** as to whether any more unprocessed data bits are left in the buffer." **(US 2005/0108312 A1 p. 12, par. 90, lines 14-15; Fig. 8C, 822)** "If the determination at block **822** is that unprocessed bits are available, the flow proceeds to tab B **826** of the flowchart **840** in **Fig. 8C**." **(US 2005/0108312 A1 p. 12, par. 92, lines 1-3; Fig. 8C, 822)**

performing a shift merge operation[3] responsive to a shift merge instruction specifying the first data block, a second data block and a shift count[4], the shift merge operation to merge said unprocessed data bits from said first data block with said second data block, wherein a merged data block is formed[5].

---

[3] "Note that the shift merge operation **740** caused the doubleword B **712** from buffer **720** to shift to the upper portion of the destination buffer **744** and doubleword C **713** was shifted into the lower portion, while the two portions were properly merged together at the middle." **(US 2005/0108312 A1 p. 11, par. 87, lines 13-19; Fig. 7B, 740)** "For this situation, the next symbol at block **820** is the merged result of the partial symbol bits from block **842** and the corresponding partial symbol bits that were loaded with the bitstream data from block **844** " **(US 2005/0108312 A1 p. 12, par. 92, lines 23-26; Fig. 8C, 844)** See also **pp. 9-10, pars. 75-81; Figs. 5A-B & 6A-B.**

[4] "The PSRMRG instruction for a shift right merge (also, a register shift) operation of this embodiment begins with three pieces of information: a first data operand **402**, a second data operand **404**, and a shift count **406**. In one embodiment, the PSRMRG shift merge instruction is decoded into one micro-operation. In an alternate embodiment, the instruction may be decoded into a various number of micro-ops to perform the shift merge operation on the data operands. For this example, the data operands **402**, **404**, are 64 bit wide pieces of data stored in a register/memory and the shift count **406** is an 8 bit wide immediate value. Depending on the particular implementation, the data operands and shift count can be other widths such as 128/256 bits and 16 bits, respectively. The first operand **402** in this example is comprised of eight data segments: P, O, N, M, L, K, J, and I. The second operand **404** is also comprised of eight data segments: H, G, F, E, D, C, B, and A. The data segments here are of equal length and each comprise of a single byte (8 bits) of data. However, another embodiment of the present invention operates with longer 128 bit operands wherein the data segments are comprised of a single byte (8 bits) each and the 128 bit wide operand would have sixteen byte wide data segments." **(US 2005/0108312 A1 p. 8, par. 69, lines 3-25; Figs. 4A-B, 404, 402 & 406 respectively)** "For these discussions, MM1 **504**, MM2 **506**, TEMP **532**, and DEST **542**, are generally referred to as operands or data blocks, but are not restricted as such and also include registers, register files, and memory locations. In one embodiment MM1 **504** and MM2 **506** are 64 bits wide MMX registers (also referred to as 'mm' in some instances). At the state I **500**, a shift count imm[*y*] **502**, a first operand MM1[*x*] **504**, and a second operand MM2[*x*] **506** are sent with the parallel shift right merge instruction. The count **502** is an immediate value of *y* bits width. The first **504** and second **506** operands are data blocks including *x* data segments and having total widths of 8*x* bits each if each data segment is a byte (8 bits). The first **504** and second **506** operands are each packed with a number of smaller data segments. For this example, the first data operand MM1 **504** is comprised of eight equal length data segments: P **511**, O **512**, N **513**, M **514**, L **515**, K **516**, J **517**, I **518**. Similarly, the second data operand MM2 **506** is comprised of eight equal length data segments: H **521**, G **522**, F **523**, E **524**, D **1225**, C **526**, B **527**, A **528**. Thus each of these data segments are '*x* * 8' bits wide. So if *x* is 8, each operand is 8 bytes or 64 bits wide. For other embodiments, a data element can be a nibble (4 bits), word (16 bits), double word (32 bits), quad word (64 bits), etc. In alternate embodiments, *x* can be 16, 32, 64, etc. data elements wide. The count *y* is equal to 8 for this embodiment and the immediate can be represented as a byte. For alternate embodiments, *y* can be 4, 16, 32, etc. bits wide. Furthermore, the count **502** is not limited to an immediate value and can also be stored in a register or memory location." **(US 2005/0108312 A1 p. 9, par. 75, lines 3-32; Figs. 5A-B, 506, 504 & 502 respectively)**

[5] "The right shift logic **414** outputs the appropriate number of data segments from the temporary register as the resultant **408**. In another embodiment, an output multiplexer or latch can be included after the right shift logic to output the resultant. For this example, the resultant is 64 bits wide and includes eight bytes. Due to the shift right merge operation on the two data operands **402**, **404**, the resultant is comprised of the following eight data segments: K, J, I, H, G, F, E, and D." **(US 2005/0108312 A1 p. 8, par. 71, lines 22-30; Figs. 4A-B, 408)** "At state III **540**, the data segments (K, J, I, H, G, F, E, D) framed by the window

Similarly, claim 24 sets forth an article comprising a tangible machine readable medium that stores a program[6], said program being executable by a machine to perform a method comprising: determining whether any unprocessed data bits for a partial variable length symbol exist[1] in a first data block[2]; and performing a shift merge operation[3] responsive to a shift merge instruction specifying the first data block, a second data block and a shift count[4], the shift merge operation to merge said unprocessed data bits from said first data block with said second data block, wherein a merged data block is formed[5].

Claim 15 sets forth an apparatus comprising: an execution unit[7] to execute a plurality of instructions for a variable length decoding algorithm, wherein one of said instructions is a first instruction for a shift merge operation[3], said plurality of instructions to cause said execution unit to: determine whether any unprocessed data bits for a partial variable length symbol exist[1] in a first data block[2]; and perform a shift merge operation[3] responsive to a shift merge instruction specifying the first data block, a second data block and a shift count[4], the shift merge operation to merge said unprocessed data bits from said first data block with said second data block, wherein a merged data block is formed[5].

---

534 is outputted as a resultant to an *x* data elements wide destination DEST[*x*] **542**." **(US 2005/0108312 A1 pp. 9-10, par. 76, lines 29-31; Figs. 5A-B, 542)**

[6] "In one embodiment, the methods of the present invention are embodied in machine-executable instructions. The instructions can be used to cause a general-purpose or special-purpose processor that is programmed with the instructions to perform the steps of the present invention. The present invention may be provided as a computer program product or software which may include a machine or computer-readable medium having stored thereon instructions which may be used to program a computer (or other electronic devices) to perform a process according to the present invention." **(US 2005/0108312 A1 p. 2, par. 28, lines 4-15)**

[7] "The data operands **402**, **404**, and the count **406** are sent to an execution unit **410** in the processor along with a shift right merge instruction. By the time the shift right merge instruction reaches the execution unit **410**, the instruction should have been decoded earlier in the processor pipeline. Thus the shift right merge instruction can be in the form of a micro operation (uop) or some other decoded format." **(US 2005/0108312 A1 p. 8, par. 70, lines 2-9; Fig. 4A, 410)** "The data operands **402**, **404**, and the count **406** are sent to an execution unit **420** in the processor along with a shift right merge instruction. For this embodiment, the first data operand **402** and the second data operand **404** are received at shift left logic **422** and shift right logic **424**, respectively. The count **406** is also sent to the shift logic **422**, **424**." **(US 2005/0108312 A1 p. 9, par. 73, lines 1-7; Fig. 4B, 420)**

VI. Grounds of Rejection to be Reviewed on Appeal

A. Claims **1**-14, **15**-23 and **24**-29 stand rejected under 35 USC § 101 as allegedly being directed to a program per se.

VII. Argument

A. 35 U.S.C. § 101 REJECTIONS

Claims **1**-14, **15**-23 and **24**-29 stand rejected under 35 USC § 101 as allegedly being directed to a program per se, because no computer-readable medium is recited for recording functionally descriptive material.

Appellant respectfully disagrees.

Claim 24 clearly sets forth such a medium, for example:

> 1. (Previously Presented) An article comprising a tangible machine readable medium that stores a program, said program being executable by a machine to perform a method comprising:
> determining whether any unprocessed data bits for a partial variable length symbol exist in a first data block; and
> performing a shift merge operation responsive to a shift merge instruction specifying the first data block, a second data block and a shift count, the shift merge operation to merge said unprocessed data bits from said first data block with said second data block, wherein a merged data block is formed.

Even further, with regard to process claims 1-14, appellant respectfully submits that the examiner is in error for concluding that they are directed to a program per se. Claim 1 sets forth:

> 1. (Previously Presented) A method comprising:
> determining whether any unprocessed data bits for a partial variable length symbol exist in a first data block; and
> performing a shift merge operation responsive to a shift merge instruction specifying the first data block, a second data block and a shift count, the shift merge operation to merge said unprocessed data bits from said first data block with said second data block, wherein a merged data block is formed.

An analysis of the instant claims must be performed in order to make a determination of whether the subject matter is statutory. Such analysis should correlate each claim element with corresponding structures, materials or acts set forth in the specification.

For example, page 2, par. 28, of the specification (emphasis added) discloses that:

> In one embodiment, the methods of the present invention are embodied in machine-executable instructions. The instructions can be used to cause a general-purpose or special-purpose processor that is programmed with the instructions to perform the steps of the present invention. The present invention may be provided as a computer program product or software which may include a machine or computer-readable medium having stored thereon instructions which may be used to program a computer (or other electronic devices) to perform a process according to the present invention. Alternatively, the steps of the present invention might be performed by specific hardware components that contain hardwired logic for performing the steps, or by any combination of programmed computer components and custom hardware components.

Therefore, appellant respectfully submits that the examiner is in error for miscategorizing the instant claim as a mere program listing rather than as a statutory process or improvement thereof.

Appellant respectfully submits that, following the examiner's reasoning, any process that could be automated through programmed machines would be non-statutory as a program per se. Indeed it may very well be the case that any process which can be listed as steps in a claim can in fact be automated by programmed machines. Therefore, according to the examiner's reasoning, any process which can be listed as steps in a claim is not patentable.

To this point, the Federal Circuit explained in *AT & T Corp. v. Excel Communications, Inc.* 172 F.3d 1352, 1356, 50 USPQ2D 1447, 1450 (Fed. Cir. 1999) that (emphasis added):

> This court recently pointed out that any step-by-step process, be it electronic, chemical, or mechanical, involves an "algorithm" in the broad sense of the term. *See State Street Bank & Trust Co. v. Signature Fin. Group, Inc.*, 149 F.3d 1368, 1374-75, 47 USPQ2d 1596, 1602 (Fed. Cir. 1998).

Yet 35 U.S.C. §101 clearly sets forth that (emphasis added), "Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title."

Thus, contrary to the examiner's assertion, the intent of such broad coverage for "any new and useful process" does not prohibit some particular new and useful processes simply because they can be performed by, or can make use of a computer program.

Therefore, appellant respectfully submits that the Examiner is in error for incorrectly concluding that the process of claims 1, the executable program on a tangible machine readable medium of claim 24, and the programmable apparatus of claim 15 are each directed to programs per se (i.e. programs—of, in or by themselves, their functionally descriptive material standing alone without reference to additional means which would allow said functionality to be realized). Indeed, the MPEP §2106.01 further states that (emphasis added):

> Computer programs are often recited as part of a claim. USPTO personnel should determine whether the computer program is being claimed as part of an otherwise statutory manufacture or machine. In such a case, the claim remains statutory irrespective of the fact that a computer program is included in the claim. The same result occurs when a computer program is used in a computerized process where the computer executes the instructions set forth in the computer program. Only when the claimed invention taken as a whole is directed to a mere program listing, i.e., to only its description or expression, is it descriptive material *per se* and hence nonstatutory.

Since the instant claims are not directed to a mere program listing but rather to otherwise statutory processes, articles and apparatus, the question at hand, is whether such processes, articles and apparatus are "new and useful."

The Federal Circuit makes it clear that the ordinary and customary meaning of a claim term is the meaning that the term would have to a person of ordinary skill in the art in question at the time of the invention. "The person of ordinary skill in the art is deemed to read the claim term not only in the context of the particular claim in which the disputed term appears, but in the context of the entire patent, including the specification." *Phillips v. AWH Corp.*, 415 F.3d at 1313.

Appellant respectfully submits that the instant language, when correlated with the corresponding structures and processes set forth in the specification makes it apparent to one of skill in the art that the claimed invention has a practical application in the technical arts, i.e. to improve the performance of variable length encoding (VLC) and decoding (VLD) such as are found in image and video processing tasks, and in communications, and in various compression techniques and standards, e.g. JPEG and MPEG.

In addition, appellant respectfully submits, that the present application clearly asserts such a practical application in the technical arts.

For example, paragraphs 33-34 of the specification (emphasis added) assert that:

> Applications of coding and decoding operations are found in a wider array of image and video processing tasks and communications. One example of coding/decoding algorithms is used in processing of Motion Picture Expert Group (MPEG) video. Variable length encoding (VLC) and decoding (VLD) are used in various compression techniques and standards such as JPEG and MPEG. In variable length codes, the different symbols require a different numbers of bits. One operation in variable length decoding is to extract bits out of the bitstream before decoding the symbols. In order to extract a variable number of bits from the bitstream for a variable length symbol, the beginning of symbols have to be addressed by bits instead of bytes. However, addressing memory on a bit-wise level is difficult, data from the bitstream is sent to a temporary register first. The bits are then shifted around and manipulated to emulate bit addressability of the bitstream.
>
> Unfortunately, current methods and instructions target the general needs of variable length coding/decoding and are not comprehensive. In fact, many architectures do not support a means for efficient extraction of varying length data symbols. In addition, data ordering within data storage devices such as SIMD registers, as well as a capability of merging values in a register and for partial data transfers between registers, are generally not supported. As a

> result, current architectures require unnecessary data type changes which increases the number of clock cycles required to order data for arithmetic operations. A SIMD shift merge instruction can be useful in audio and video applications where large amounts of packed data are processed. For example, a single shift merge instruction of one embodiment is capable of replacing multiple instructions that would be needed to perform an equivalent data manipulation. By reducing the number of instructions needed, throughput can be increased and processing resources such as registers and execution units freed up.

Thus the specification makes it readily apparent to one of skill in the art that the claimed invention has a practical application in the technical arts.

The Supreme Court held that the focus in any statutory subject matter analysis be on the claim as a whole, stating "When a claim containing a mathematical formula implements or applies that formula in a structure or process which, when considered as a whole, is performing a function which the patent laws were designed to protect (e.g., transforming or reducing an article to a different state or thing, then the claim satisfies the requirements of § 101." *In re Alappat*, 33 F.3d 1526, 1543 (Fed. Cir. 1994) (quoting *Diehr*, 450 U.S. at 192, 209 USPQ at 10).

The final Office Action (p. 2, par. 4) further asserts that the instant claims are non-statutory for being directed to a data transformation, rather than a physical transformation.

This notion has sometimes been phrased in terms of requiring a transformation or reduction of 'subject matter.' In Schrader, the phrase 'subject matter' was determined not to be limited to tangible articles or objects, but includes intangible subject matter, such as data or signals (e.g. in digital images, video and communications), representative of or constituting physical activity or objects. *Schrader*, 22 F.3d at 295, 30 USPQ2D (BNA) at 1459.

Therefore, since data of digital images, video and communications being transformed is representative of or constituting physical activity or objects, appellant

respectfully submits that Claims **1**-14, **15**-23 and **24**-29 are directed to statutory subject matter.

Conclusion

Appellant submits that all claims now pending are in condition for allowance. Such action is earnestly solicited at the earliest possible date. If there is a deficiency in fees, please charge our Deposit Acct. No. 50-0221.

Respectfully submitted,

Date: September 11, 2007

/s/Lawrence M. Mennemeier/
Lawrence M. Mennemeier
Reg. No. 51,003

INTEL CORPORATION
c/o INTELLEVATE LLP
P.O. Box 52050
Minneapolis, MN 55402
(408) 765-2194

VIII. Claims Appendix: Claims Allowed and Involved in Appeal (Clean Copy)

1. (Previously Presented) A method comprising:

determining whether any unprocessed data bits for a partial variable length symbol exist in a first data block; and

performing a shift merge operation responsive to a shift merge instruction specifying the first data block, a second data block and a shift count, the shift merge operation to merge said unprocessed data bits from said first data block with said second data block, wherein a merged data block is formed.

2. (Original) The method of claim 1 further comprising receiving said first data block and said second data block from a bitstream.

3. (Original) The method of claim 1 wherein said first data block and said second data block are loaded from bitstream data in memory.

4. (Original) The method of claim 2 further comprising extracting a merged variable length symbol from said merged data block, said merged variable length symbol comprised of said unprocessed data bits and a plurality of data bits from said second data block.

5. (Original) The method of claim 4 wherein said merged variable length symbol is sent

to a register.

6. (Previously Presented) The method of claim 4 further comprising:

evaluating said merged data block for variable length symbols;

extracting any whole variable length symbols located in said second data block; and

sending any extracted whole variable length symbols to a register.

7. (Original) The method of claim 6 wherein each of said variable length symbols is comprised of at least N-bits of data.

8. (Original) The method of claim 7 wherein N is a number greater than one.

9. (Original) The method of claim 2 wherein said first data block is held in a first single instruction multiple data register.

10. (Previously Presented) The method of claim 9 wherein said first single instruction multiple data register is a buffer.

11. (Original) The method of claim 2 wherein said second data block is held in a second single instruction multiple data register.

12. (Previously Presented) The method of claim 1 wherein said shift merge operation is a single instruction multiple data type of instruction to cause a parallel shift right

merge of data operands based on the specified shift count.

13. (Original) The method of claim 10 wherein said shift merge operation operates on data elements at a byte granularity.

14. (Original) The method of clam 10 wherein said shift merge operation operates on data elements at a bit granularity.

15. (Previously Presented) An apparatus comprising:

an execution unit to execute a plurality of instructions for a variable length decoding algorithm, wherein one of said instructions is a first instruction for a shift merge operation, said plurality of instructions to cause said execution unit to:

determine whether any unprocessed data bits for a partial variable length symbol exist in a first data block; and

perform a shift merge operation responsive to a shift merge instruction specifying the first data block, a second data block and a shift count, the shift merge operation to merge said unprocessed data bits from said first data block with said second data block, wherein a merged data block is formed.

16. (Original) The apparatus of claim 15 wherein plurality of instructions further cause said execution unit to extract a merged variable length symbol from said merged data block, said merged variable length symbol comprised of said unprocessed data bits and a plurality of data bits from said second data block.

17. (Original) The apparatus of claim 15 wherein said first data block and said second data block are received from a bitstream.

18. (Original) The apparatus of claim 16 wherein said plurality of instructions further cause said execution unit to: evaluate said merged data block for variable length symbols; and extract any whole variable length symbols located in said second data block.

19. (Original) The apparatus of claim 18 wherein each of said variable length symbols is comprised of at least two bits of data.

20. (Original) The apparatus of claim 19 wherein said first data block and said second data block are held in a first single instruction multiple data register and a second single instruction multiple data register, respectively.

21. (Original) The apparatus of claim 20 wherein said first single instruction multiple data register is a buffer.

22. (Previously Presented) The apparatus of claim 21 wherein said shift merge operation is a single instruction multiple data type of instruction to cause a parallel shift right merge of data operands based on the specified shift count.

23. (Original) The apparatus of claim 22 wherein said shift merge operation is to operate

on data elements at a byte granularity.

24. (Previously Presented) An article comprising a tangible machine readable medium that stores a program, said program being executable by a machine to perform a method comprising:

determining whether any unprocessed data bits for a partial variable length symbol exist in a first data block; and

performing a shift merge operation responsive to a shift merge instruction specifying the first data block, a second data block and a shift count, the shift merge operation to merge said unprocessed data bits from said first data block with said second data block, wherein a merged data block is formed.

25. (Original) The article of claim 24 wherein said first data block and said second data block are to be loaded from a data bitstream.

26. (Original) The article of claim 25 wherein said method further comprises extracting a merged variable length symbol from said merged data block, said merged variable length symbol comprised of said unprocessed data bits and a plurality of data bits from said second data block.

27. (Original) The article of claim 26 wherein each of said variable length symbols is comprised of at least two bits of data.

28. (Previously Presented) The article of claim 24 wherein said shift merge operation is a single instruction multiple data type of instruction to cause a parallel shift right merge of data operands based on the specified shift count.

29. (Original) The article of claim 26 wherein said method further comprises: evaluating said merged data block for variable length symbols; and extracting any whole variable length symbols located in said second data block.

30. (Previously Presented) A system comprising: a memory to store data and instructions, a processor coupled to said memory on a bus, said processor operable to perform instructions for a variable length decoding algorithm, said processor comprising:

a bus unit to receive a sequence of instructions from said memory;

an execution unit coupled to said bus unit, said execution unit to execute said sequence, said sequence to include a first instruction specifying a first data block, a second data block and a shift count for a shift merge operation, said sequence to cause said execution unit to:

determine whether any unprocessed data bits for a partial variable length symbol exist in the first data block; and

perform the shift merge operation responsive to the first instruction to merge said unprocessed data bits from said first data block with said second data block, wherein a merged data block is formed.

31. (Original) The system of claim 30 wherein said first data block and said second data

block are loaded from bitstream data.

32. (Original) The system of claim 30 wherein said plurality of instructions further cause said execution unit to extract a merged variable length symbol from said merged data block, said merged variable length symbol comprised of said unprocessed data bits and a plurality of data bits from said second data block.

33. (Original) The system of claim 32 wherein said plurality of instructions further cause said execution unit to: evaluate said merged data block for variable length symbols; and extract any whole variable length symbols located in said second data block.

34. (Original) The system of claim 33 wherein each of said variable length symbols is comprised of at least two bits of data.

35. (Original) The system of claim 34 wherein said first data block and said second data block are held in a first single instruction multiple data register and a second single instruction multiple data register, respectively.

36. (Previously Presented) The system of claim 35 wherein said shift merge operation is a single instruction multiple data type of instruction to cause a parallel shift right merge of data operands based on the specified shift count.

37. (Original) The system of claim 36 wherein at least a single edge for at least one or

more of said variable length symbols is not aligned at a byte boundary.

38. (Original) The system of claim 37 wherein said shift merge operation is to operate on data elements at a bit granularity.

39. (Original) The system of claim 37 wherein said shift merge operation is to operate on data elements at a byte granularity.

IX. Evidence Appendix: With Copies of Evidence Relied Upon by Appellant

Appellant relies upon no additional evidence in this appeal.

X. Related Proceedings Appendix: Copies of Decisions Rendered by a Court or the Board in any Prior and Pending Appeals, Interferences or Judicial Proceedings

There are no related appeals or interferences to appellant's knowledge that would have a bearing on any decision of the Board of Patent Appeals and Interferences.